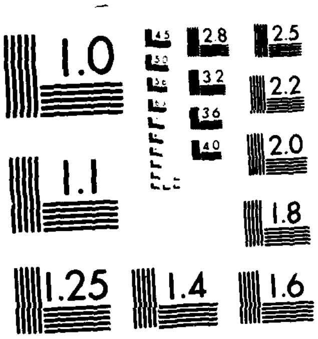

1.0
4.5
5.0
5.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

②

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
AIM 683

2. GOVT ACCESSION NO.
A127251

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Visual Algorithms

5. TYPE OF REPORT & PERIOD COVERED
Memorandum

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Tomas Lozano-Perez

8. CONTRACT OR GRANT NUMBER(s)
N00014-75-C-0643
MCS-79-23110
Nato Grant NO. 237.81

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Artificial Intelligence Laboratory
545 Technology Square
Cambridge, Massachusetts 02139

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Advanced Research Projects Agency
1400 Wilson Blvd
Arlington, Virginia 22209

12. REPORT DATE
May 1982

13. NUMBER OF PAGES
28

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
Office of Naval Research
Information Systems
Arlington, Virginia 22217

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Polynomial Algorithms
Parallel/Serial
Neuronal Hardware
Perceptions
Nonlinear Mappings

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Nonlinear, local and highly parallel algorithms can perform serveral simple but important visual computations. Specific classes of algorithms can be consideredin an abstract way. I study here the class of polynomial algorithms to exemplify some of the important issues for visual processing like linear vs nonlinear and local vs. global. Polynomial algorithms are a natural extension o Perceptrons to time dependent grey level images. Although they share most of the limitations of Perceptrons, they are powerful parallel computational devic
CON'T

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Several of their properties are characterized and especially (a) their equivalence with Perceptrons for geometrical figures and (b) the synthesis of nonlinear algorithms (mappings) via associative learning. Finally, the paper considers how algorithms of this type could be implemented in nervious hardware, in termsof synaptic interactions strategically located in a dendritic tree. The implementaion of three specific algorithms is briefly outlined:

(a) direction sensitive motion detection
(b) detection of discontinuities in the optical flow
(c) detection and localization of zero-crossings in the convulution of the imag with the Laplacian (of a Gaussian). In the appendix, another(nonlinear) differential operator, the second directional derivative along the gradient, is briefly discussed as an alternative to the Laplacian.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 683 May, 1982

VISUAL ALGORITHMS

Tomaso Poggio

Abstract: Nonlinear, local and highly parallel algorithms can perform several simple but important visual computations. Specific classes of algorithms can be considered in an abstract way. ~~I study~~ here the class of polynomial algorithms to exemplify some of the important issues for visual processing like linear vs. nonlinear and local vs. global. Polynomial algorithms are a natural extension of Perceptrons to time dependent grey level images . Although they share most of the limitations of Perceptrons, they are powerful parallel computational devices. Several of their properties are characterized and especially ~~(a)~~ their equivalence with Perceptrons for geometrical figures and ~~(b)~~ the synthesis of nonlinear algorithms (mappings) *via* associative learning. Finally, the paper considers how algorithms of this type could be implemented in nervous hardware, in terms of synaptic interactions strategically located in a dendritic tree. The implementation of three specific algorithms is briefly outlined:

~~(a)~~ direction sensitive motion detection;

~~(b)~~ detection of discontinuities in the optical flow, and

~~(c)~~ detection and localization of zero-crossings in the convolution of the image with the Laplacian (of a Gaussian). In the appendix, another (nonlinear) differential operator, the second directional derivative along the gradient, is briefly discussed as an alternative to the Laplacian.

Acknowledgement. This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology . Support for the Laboratory's artificial intelligence research is provided, by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-75-C-0643; this work was supported in part by National Science Foundation Grant MCS-79-23110 and travel for collaboration purposes was supported under NATO grant number 237.81.

1. EARLY VISUAL ALGORITHMS

1.1. Algorithms Depend on Computation and Hardware

One can distinguish (or at least I did so with David Marr:see Marr and Poggio, 1976; Marr, 1982) at least three levels at which a visual processor must be understood. At the top level is the computational theory of the device in which the problem to be solved is characterized, and the natural constraints are made explicit. At the bottom is the level of the detailed neuronal "hardware" - neural circuits, synapses and so forth - that perform the computation. In the middle is a study of the algorithms used to compute the solution. This second level is the hardest to define precisely since it represents a bridge between the computational level and the hardware level. Thus, while the circuitry is determined by the available mechanisms and the computation by the nature of the problem, the algorithm itself is determined by the computation *and* by the available hardware.

David Marr has especially stressed the computational level of analysis since it is a level of explanation which is still new to neurobiology. Together we have stressed that the relationships between these levels are rather loose. In this paper I find it especially appropriate to emphasize that it is hopeless to understand the algorithms used by a biological or artificial processor without knowing which computational problem is solved and what are the properties and the limitations of the hardware.

Both the mechanisms and the problem provide powerful constraints to the possible algorithms. Horace Barlow made this point very clearly when, in his Ferrier lecture (1981), he spoke about the "limiting requirements" imposed by the physics of light, i.e., the nature of the visual world, and the properties of the nervous mechanisms, for instance the limited precision of the connections and the noise of nervous transduction. For instance, the von Neuman architecture of classical computers depends almost entirely on the type of available processing elements which made concurrency cumbersome to implement. In the nervous system the processing elements - neurons and synapses, as I will discuss later - are abundant and flexible. VLSI is now bringing similar advantages to circuits. Connections, however, are still vastly more numerous and more flexible in the brain than in solid state electronics, where they are restricted to 2-D surfaces. The costs of internal communication are still exorbitant in today's computers.

It is therefore not surprising that algorithms strongly depend on the constraints imposed by the hardware. I would argue that the main reason for the large gap presently existing between computational theories and computer scientists on one end and physiology on the other end is our ignorance of the nature and the properties of the biological hardware performing the elementary steps of information processing. Biophysics of information processing, which I will discuss later, is as necessary for analyzing and understanding the algorithms used by the brain as the computational analysis of the specific tasks.

1.2. Systems and Algorithms

Visual information processing begins with a large array of photoreceptors that transduce local light intensities into time dependent signals. The information about the outside world and how it changes is implicit in this retinotopic array of signals and must be decoded by a variety of processes or algorithms. Formally these algorithms, considered as "black boxes", have many inputs the photoreceptors and are in general nonlinear.

Since I restrict my discussion to the first steps of visual information processing, I will consider algorithms that operate almost directly on the photoreceptor signals, i.e., on the primary intensity representation (Nishihara, 1981). At this level, and especially for biological systems, it is natural to treat algorithms as systems or operators mapping a set of inputs into a set of outputs. From this point of view, two simple dichotomies can characterize, albeit rather superficially, early visual algorithms: 1) linear vs. nonlinear; and 2) parallel vs. serial.

Systems have inputs and outputs. Mathematically a system is equivalent to an operator which maps functions into functions (in a suitable space). An operator can be defined in at least two ways: (a) as a catalogue of all the inputs and the corresponding outputs; and (b) as an algorithm, i.e., an explicit law or set of rules that enables one to compute the output for any given input.

These two descriptions are met under different forms in various contexts. In information theory a view close to (a) leads to the classical combinatorial definition of information measure, while an algorithmic view leads to the Kolmogorov's definition of information entropy. In computer technology logical operations are often performed through a look-up table, i.e., a catalogue.

1.3. Linear vs. Nonlinear

Nonlinear systems represent in the space of all systems a much larger class than linear systems. The restriction of linearity is very strong and sets powerful general constraints on the system's behavior and properties. A linear system is a map L satisfying

$$L(ax) = aL(x)$$

$$L(x_1 + x_2) = L(x_1) + L(x_2)$$

From the information processing point of view, the limitations of linear systems are clear. Linear operations cannot perform conjunctions and discriminate the intersection of events. In a sense which can be made more precise multiplications or divisions are necessary to provide a sufficiently powerful set of basic operations. The crucial processes of an information processing device require logical operations that are essentially nonlinear and more like multiplications than addition or subtraction.

For a computer scientist this question of linear vs. nonlinear algorithms may indeed seem a straw problem: after all every computing machine is intrinsically nonlinear, is full of nonlinearities. For neurobiologists, however, the question of linearity vs. nonlinearity of some nervous subsystem - and of the operation thereby implemented - is non trivial. As we will see later, the classical view of the neuron and the concept of integrative action may be flawed by the failure to recognize the necessity of nonlinear operation on graded synaptic inputs.

1.4. Parallel vs. Serial

The distinction between parallel vs. serial processing is almost a commonplace in a variety of different areas (of course the concepts of parallel and serial processing have a range of meanings). It is, for instance, the most frequently cited difference between present computers and brains. The amount and the nature of wiring are vastly different: it is easy to create many very small transistors with present solid state technologies but more difficult to produce extensive connections among them. In a brain each nervous cell receives thousands of inputs. Nervous wiring is not restricted to two-dimensional surfaces. Especially in the first parts of the visual pathway nervous processing is indeed undoubtedly spatially parallel and preserving the topography of the image (at least up to area 17 and other visual areas, modulo a conformal mapping that preserves local geometry).

Algorithms with a more serial flavor are certainly used at later stages in the nervous system. Even in vision, however, serial processing is likely to play an important role quite early on, certainly earlier than most neurobiologists accustomed to the idea of topographic maps and inner screen would be ready to admit.

From a computational point of view the most interesting issues about parallel algorithms revolve around the notion of "local and global" or "parts and wholes". Loosely speaking, a computational problem is inherently local if it can be divided into small, non-interacting modules. It is inherently global if any way of dividing it into subcomponents must entail substantial interaction among the modules.

1.5. Plan of the paper

It is, however, very difficult to characterize the locality or globality, as well as the linearity or nonlinearity, of a computational task without any reference to a specific and possibly abstract class of machines on which the computation will run. I will thus consider a class of algorithms and related abstract machines that are a natural extension of Perceptrons and briefly formalize in this framework the issue of *local and global* as well as of *linear and nonlinear*. The motivations for considering this particular class of algorithms is more fully discussed by Poggio and Reichardt (1980). The main attraction of polynomial algorithms is their generality: they approximate, under rather weak conditions, all *smooth* input-output transductions. Heuristically, several early visual computations seem indeed to have this smooth character of transducing input signals continuously into output functions, without major discontinuities or "decisions". The main results on polynomial algorithms are summarized in section 2., mainly from Poggio and Reichardt (1980), to which we refer the reader for detailed definitions, proofs and references. Some of the results on coding of the input set (section 2.8) and nonlinear associative mappings (section 2.9 and 2.10) are new. In the context of this paper the main thrust of the next section is to show formally that several simple but important computations require local (i.e. highly parallel) algorithms with nonlinearities of a simple kind. The question is then how could these interactions be implemented in neuronal hardware? Section 3. suggests that a specific biophysical mechanism may perform the key operation in many local, nonlinear visual algorithms. The section is a brief and incomplete summary of several recent papers. Three examples of specific algorithms based on this mechanism are proposed in the last section for (a) detecting directional motion, (b) separating figure from ground and (c) localizing zero-crossings . It is conjectured that the corresponding circuitries may indeed be used by some biological visual systems. The first two example are summarized from Torre and Poggio (1978) and Poggio et al. (1981a), respectively. The neuronal circuitry for the detection of zero-crossings is original. It may be useful to point out that this paper is not fully selfcontained. Its main purpose is rather to outline several recent developments and to connect them in a new and coherent framework: the many gaps and missing details should be filled in from the original papers.

2. POLYNOMIAL ALGORITHMS

As I mentioned, the input space relevant for us is a 2-D array of time-dependent signals. We formalize this, defining as *retina* the collection of N photoreceptors arranged on a 2-D lattice, and by a pattern on the retina a set of input functions, seen by the photoreceptors. Then a polynomial algorithm on the pattern X is a mapping to an output function with the form

$$S[x] = \sum_{i=1}^{N} \sum_{j_1 \ldots j_i} L_{j_1 \ldots j_i}[x(t)] \qquad (1)$$

where $L_{j_1 \ldots j_i}$ is an i-linear form in the i components of the input array $[x_1(t) \ldots x_N(t)]$. Eq. 1 is a natural extension of the usual algebraic polynomials: inputs and outputs are here *functions* instead of real (or complex) numbers.

2.1. Inputs

When $x_i(t) = x_i$, the input pattern is a grey-level "figure" on the retina. If the x_i take only 0 and 1 values the pattern is essentially a "geometric figure" as considered in "Perceptrons" by Minsky and Papert (1969). Since a linear transformation of the pattern does not change the type of representation (eq. 1), it is often useful to think of the pattern as a (linearly!) filtered version of the brightness array (for instance through a Difference of Gaussians operator, a DOG).

2.2. Graphs

Eq. 1 is equivalent to the decomposition of the operator S into the sum of interactions of different orders between the input functions. Each term can be represented by a certain graph and thus an N-input system can be decomposed into a sequence of graphs (see Fig. 1). The graphs are actually another notation for the polynomial operator itself. In particular, composition of systems can be computed directly in terms of the graphical notation.

2.3. Three Questions

Three interesting questions can be asked about these operators. The first one concerns the existence of an explicit representation. The second problem is how wide is the class of algorithms that can be approximated by polynomial systems? Finally, we would like to characterize the computational properties and limitations of polynomial algorithms, especially in the framework of the local-global and linear-nonlinear dichotomies.

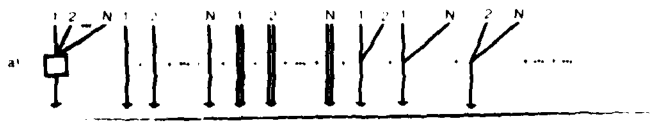


Figure 1. *The graphic representation for equation 1*

2.4. Representation and Approximation

(a) The answer to the first question (see Poggio and Reichardt 1980; Palm 1978, 1979) can be stated in terms of the following

PROPOSITION 1 (see for precise versions of this theorem Palm and Poggio 1977; Palm 1978).

All polynomial systems can be represented by symbolic integrals. Time invariant polynomial systems can be represented by Volterra series (with the kernels being distributions).

Thus the class of Volterra series coincides with the class of polynomial functionals and a polynomial algorithm (or its graph) can be written in terms of integrals and associated kernels. For instance, a linear time-invariant system can be written as a convolution

$$L_1[x(t)] = \int K(t-\tau)x(\tau)d\tau \tag{2}$$

and a second order interaction as

$$L_{12}[x_1(t), x_2(t)] = \int\int K_{12}(t-\tau_1, t-\tau_2)x_1(\tau_1)x_2(\tau_2)d\tau_1 d\tau_2 \tag{3}$$

(b) The answer to the second question depends on the type of approximation, i.e. topology, that is used (see Palm and Poggio, 1978;Palm 1978, 1979; Poggio and Reichardt, 1980). For instance, under the stochastic approximation and the pointwise approximation - rather weak approximations - polynomial systems like equation 1 approximate essentially every mapping between L^2 and R, but only a subclass of continuous mappings under the uniform approximation. Stronger results clearly hold for discrete-time systems and for discrete time, finite value input spaces. For discrete time systems all different topologies are equivalent and thus all continuous systems can be approximated by polynomial systems. If the input set is finite, then any mapping can be written as a polynomial.

2.5. Computational Properties: degree and p-order

A representation like eq. 1 is essentially a canonical decomposition of the system into the sum of simpler, standard components. Computational properties of the mapping are then "additively" determined by the computational properties of the standard components, the graphs of Fig. 1. Typically one would like to know what is the "simplest" set of graphs or interactions that can perform a given computation. Intuitively, the terms in the representation of Fig. 1b (or eq. 1) become increasingly complex going from left to right. It turns out that the concept of simplicity of a graph can be formalized in terms of the notion of degree, which measures its "nonlinearity", and p-order, which measures its "locality".

DEFINITION

A canonical graph is of degree k, if it corresponds to a k-linear form.

The degree of a graph is the total number of incoming lines. Linear graphs have degree 1, quadratic ones (bilinear forms) have degree 2.

DEFINITION

A canonical graph has p-order h, if it has inputs from h distinct "photoreceptors".

DEFINITION

The degree and p-order of a polynomial system are the maximum degree and p-order in the graphs of its canonical decomposition.

Although degree, p-order, and rank (see Poggio and Reichardt, 1980) all characterize a polynomial algorithm, the p-order is probably the single most important measure of the simplicity of a graph [linear graphs have p-order 1 (and degree 1)]. The main reason is that the notion of p-order formalizes the issue of "local vs. global" for polynomial algorithms. A lower p-order system is local: the canonical subsystems make independent, nonlinear computations based on small patches of the retina. A high p-order algorithm is global: individual graphs receive inputs from many photoreceptors in the retina. One can ask, similar to "Perceptrons", whether a certain computation is of finite p-order, i.e. if it can be computed by a polynomial mapping of some fixed p-order, regardless of the size of the retina. Notice that a linear algorithm may use inputs from many photoreceptors, but its canonical representation consists of p-order 1 graphs. Linear operations on the input patterns do not change p-order (or degree) of an algorithm (see later).

2.6. Computational Properties: Standard Machines

Several abstract computational machines could be considered for a comparison, like finite state machines, McCulloch-Pitts networks, difference equations, perceptrons, etc. The comparison clearly depends on the input set (most machines are defined only for discrete inputs with a binary number of values). If we restrict ourselves to a discrete, finite set the previous discussion implies that it is always possible to synthesize a polynomial mapping that simulates exactly the behavior of any specific machine. If we relax the constraint of a finite set, polynomial mappings are less powerful than systems with infinite memory like finite state machines, difference equations, and McCulloch-Pitts networks with loops. In practice, time is always finite and thus polynomial algorithms are equivalent to these other machines. The situation, however, clearly indicates that a polynomial description, for instance, of a finite state machine may almost always be too cumbersome to be useful. These limitations of polynomial algorithms are also indicated by their equivalence with standard perceptrons (on appropriate patterns).

2.7. Polynomial Algorithms are equivalent to Perceptrons for Geometrical Figures

The input set is restricted to "geometrical figures", i.e. to the set $[0,1]^N$ where N is the number of photoreceptors. A predicate on R is a function Ψ from a figure of R to $[0,1]$. A perceptron is a predicate on the retina R of the form (Minsky and Papert, 1969)

$$\Psi(R) = [\sum a_i \phi_i(R) > \theta]$$

where [some condition] is 1 if the condition is true and 0 if it is false. The support of ϕ is the set of all photoreceptors which affect the value of ϕ, and the order of ϕ is the size of the support of ϕ. It is straightforward to prove

Lemma

Any (perceptron) ϕ function of support n can be represented exactly for all figures by a set of polynomial graphs of p-order n and degree $(2^n - 1)$

THEOREM 2

For geometrical figures on $[0,1]^N$ perceptrons of order $\leq n$ and polynomials of p-order n and degree $(2^n - 1)$ are equivalent.

Thus the results on the order of perceptrons which compute various geometrical predicates also apply to the p-order of the corresponding polynomial mappings. Fig. 2 lists some of these results.

For both perceptrons and polynomial algorithms many simple computations turn out to be nonlocal. The limitations of perceptrons carry over to polynomial algorithms, and probably hold also for more general time dependent input patterns. This is not a surprise but is especially interesting from the point of view of the globality of given computational problems in vision. The implication is that simple, parallel algorithms, not just restricted to the Perceptron machine, can easily, i.e. locally, compute a range of important predicates, but cannot compute all of them. Apparently simple computations - like connectedness, straightness - probably require different types of algorithms, perhaps more "serial". Other computations, however, such as the computation of the direction of motion and of discontinuities in the motion field, can be performed by simple parallel algorithms (see fig.3). It may be interesting to consider the various processes involved in the first stages of visual perception from this point of view. In particular, Treismann's notion of

Computation	p-order
$[X$ is a circle$]$	4
$[\vert X\vert \geqq M]$	1
$[\vert X\vert \leqq M]$	1
$[\vert X\vert = M]$	2
$[$The n-th central moment of X about the origin is $> \vartheta]$	1
All Boolean functions of 2 variables, except $x \oplus y$ and $x \equiv y$	1
$x \oplus y$	2
$x \equiv y$	2
$[X$ is convex$]$	3

Computation	p-order
Translation invariant recognition of figures	3
$[X$ is odd$]$	[illegible] R [illegible]
$\exists\, \psi_1, \psi_2$ p-order $(\psi_1) = 1$ p-order $\psi_2 = 2$, $\psi_1 \wedge \psi_2$	[illegible]
$[X$ is connected$]$	[illegible] R [illegible]
Translation invariant geometric spectra	2
ψ in context	[illegible]
$[X$ is a hollow square$]$	3
$[$One component of X is a hollow square$]$	[illegible]
$[X$ has symmetry under reflection$]$	≤ 4
$[$Pattern A is a translate of pattern $B]$	≤ 5

Fig [illegible] ***P-order of some computations on geometrical figures***

Computation	p-order	Degree
Directional movement (translation invariant)	2	2
Directional movement (translation- and rotation invariant)	1 or 2	2
Relative movement	2 or better 4	4
Absolute position of a stimulus on the retina (not invariant to any geometric group)	1	
Expansion or contraction of a texture (translation invariant)	$\geqq 4$	
First order statistics of a (timeless) texture	1	High
Second order statistics of a (timeless) texture	2	High
Expansion-concentration of a texture (around the coordinate origin)	2	2

Figure 3. ***P-order of some computations on time dependent patterns***

separable and not separable features, Julesz' concept of instantaneous perception and Barlow's idea of topographic maps may be connected to the intrinsic "locality vs. globality" (and nonlinearity) of algorithms.

2.8. Coding of The Input Set

I mentioned earlier that linear transformations of the input patterns do not change the main characteristics of a polynomial algorithm (they simply change the

	p-order	Computation	*p*-order
	4	Translation invariant recognition of	3
	1	figures	
	1	[X is odd]	(R)
	2	$\exists \psi_1, \psi_2$ *p*-order $(\psi_1) = 1$	,
ral moment of X about	1	*p*-order $\psi_2 = 2$, $\psi_1 \wedge \psi_2$	
x)]		[X is connected]	R
inctions of 2 variables,	1	Translation invariant geometric spectra	2
y and x ≡ y		ψ in context	,
	2	[X is a hollow square]	3
	2	[One component of X is a hollow square]	,
	3	[X has symmetry under reflection]	≤ 4
		[Pattern A is a translate of pattern B]	5

Figure 2. *P-order of some computations on geometrical figures*

Computation	*p*-order	Degree
Directional movement (translation invariant)	2	2
Directional movement (translation- and rotation invariant)	1 or 2	2
Relative movement	2 or better 4	4
Absolute position of a stimulus on the retina (not invariant to any geometric group)	1	
Expansion or contraction of a texture (translation invariant)	≥ 4	
First order statistics of a (timeless) texture	1	High
Second order statistics of a (timeless) texture	2	High
Expansion-concentration of a texture (around the coordinate origin)	2	2

Figure 3. *P-order of some computations on time dependent patterns*

and not separable features, Julesz' concept of instantaneous perception ›w's idea of topographic maps may be connected to the intrinsic "locality lity" (and nonlinearity) of algorithms.

ing of The Input Set

ntioned earlier that linear transformations of the input patterns do not ıe main characteristics of a polynomial algorithm (they simply change the kernel's values). More generally a transformation of the retina R_1 into R_2 by the mapping f provides for every pattern

$$X = [x_1(t)...x_n(t)]$$

another pattern

$$Y = [y_1(t), ...y_n(t)] = f[X] = [f_1[X], ..., f_m[X]]$$

Given a polynomial mapping S_2 on R_2 we define a mapping S_2 on R_2 through

$$S_1(X) = S_2(Y) = S_2(f(X))$$

The following property is self-evident (Poggio and Reichardt, 1980)

PROPERTY

If for $i = 1...m$ the support of f is at most equal to 1, then p-order$(S_1) \leq$ p-order(S_2).

Thus nonlinear scaling of each input separately does not increase the p-order of a polynomial algorithm. Coding of this type will, however, change the order and possibly decrease it. In other words, this simple input coding may make a polynomial algorithm simpler without changing its essential properties. We can then consider the class of polynomial algorithms defined by transformation of the retina of support 1. Two principles can be used to guide the choice of appropriate transformations (Resnikoff, 1975).

Principle 1

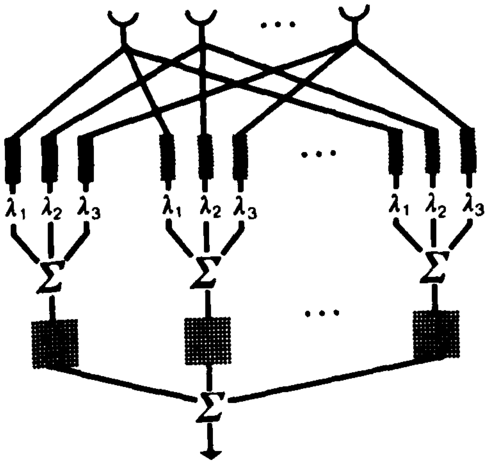
The domain and the range of the induced polynomial algorithm must coincide with the domain and range of the input-output operation (considered as function on $\mathbb{R}$*).*

Principle 2

The transformations of the retina shall be birational functions, the exponential functions and its inverse and the compositions of these functions.

Thus the transformations f_i of the retina affect the necessary conversion of domain and range with a minimal disruption of the algebraic structure of the input space . The extension from functions f_i to operators seems quite natural, when the inputs are time-dependent.

Input coding of this type simply tries to "linearize" the computation as much as possible. Input coding with support greater than 1 changes the p-order of the polynomial algorithm. Output coding also changes the properties of the algorithm. This can easily be seen in Fig.4 which shows Kolmogorov's (1963) solution of Hilbert's 13th problem: a continuous function can always be represented as the superposition of functions of 1 variable, as shown (for 3 variables) in the figure. Thus with appropriate output and input coding a p-order 1 polynomial can represent all



...
λ_1 λ_2 λ_3
λ_1 λ_2 λ_3
λ_1 λ_2 λ_3
Σ
Σ
Σ
Σ

Figure 4. *Kolmogorov's and Arnold's decomposition of a function of three variables*

continuous mappings. Interestingly, this result does not hold for continuous time functionals (Palm, 1978).

2.9. Associative Memory and Synthesis of a Polynomial Algorithm

How can one synthesize (by 'learning') a non-linear polynomial algorithm from a set of given inputs and desired outputs? In the case of a finite, discrete space of inputs the problem is well known. It is equivalent to the problem of optimal estimation of a system and to the problem of synthesizing an associative memory (Poggio, 1975 a,b). The usual problem considered in the literature concerns the synthesis of an optimal *linear* mapping M from a set of input vectors X and a set of desired output vectors Y. The best approximate mapping is given by $M = YX^+$ where X^+ is the pseudouniverse of X (see Kohonen, 1977). This technique can be extended to find the optimal polynomial mapping (Poggio, 1975 a,b). In particular, then, any mapping can be constructed in this way, at least in principle, since any mapping between two finite sets of vectors can be written as a polynomial (see earlier). This can also be derived from another interesting result, that I discuss next.

2.10. Associative Memories and Nonlinearities

The role of nonlinearities in an associative memory scheme has been long recognized as critically important. The following simple theorem provides a general connection between linear associative schemes and nonlinearities.

PROPOSITION 3

Any nonlinear associative mapping between two finite sets of vectors is equivalent to a linear associative mapping preceded by (nonlinear) input coding.

PROOF

The proof is based on the following two obvious lemmas (given, implicitly, in Palm 1978; Poggio, 1975b).

LEMMA 1 Any (nonlinear) mapping between two finite sets of vectors can be written as a polynomial.

LEMMA 2 Any polynomial mapping between two finite sets of vectors

$$Y = L_0 + L_1[X] + L_2[X, X] ...$$

is a linear mapping on appropriate crossproducts of the elements of X (i.e. on the tensor products X, $X \otimes X, ...$).

The synthesis of these crossproducts can be seen as nonlinear input coding (of support >1). [Output coding instead of input coding has similar properties (see Poggio 1975b)]. A simple but striking result follows again:

COROLLARY

Any nonlinear mapping between two finite sets of vector can be synthesized associatively with the pseudoinverse technique.

For more general input sets the problem cannot be answered exactly but only in terms of approximate associative mappings. The idea of coding an input set before performing the bulk of the computation or association is clearly powerful and can be found in a variety of contexts. In the framework of associative memory it is again interesting to notice the connection with the Kolmogorov result: input coding here does *not* depend on the mapping to be synthesized but the final "output" function g does. The result shows that for a continuous function an n-dimensional table can be replaced by a 1-dimensional table representing g and some input coding. This does not necessarily reduce the memory requirements in all cases. However, it may be conjectured that an appropriate choice of the input coding for a specific class of input-output operations may allow significant reductions in memory size (see Poggio and Rosser, 1982).

In summary, polynomial algorithms are powerful parallel computational devices and it is important to stress not only their limitations but also their locality in a number of important computations. They may be useful for characterizing simple parallel processing operations in a visual system. The detection of motion and relative motion can be characterized in terms of simple polynomial algorithms and general properties can be proved for a whole range of specific models. As discussed earlier, however, the algorithms used by a system are strongly constrained by the available hardware. In the next section I will briefly discuss a class of mechanisms - local interactions between synaptic inputs - possibly used by the nervous systems. I will also show that these interactions compute in fact specific polynomial functionals of the inputs. Thus the link between polynomial algorithms and nervous hardware may turn out to be, at least in some instances, rather direct.

3. BIOPHYSICS OF INFORMATION PROCESSING

Many different types of algorithms could be used in early vision. The computational problem does not provide sufficient constraints to uniquely define the algorithm. Even properties like local vs. global may vary for the same computation between different classes of algorithms. Ultimately the hardware of the computer or of the brain imposes critical limiting factors that constrain the class of algorithms. What is then the hardware of the brain? Where are the elementary operations performed and what do they look like?

The traditional view is that the threshold mechanism associated with spike generation performs the elementary logical operations: a neuron fires if the sum of its inputs exceeds a certain threshold and is otherwise silent. All logical operations can be implemented in this way, via McCullogh-Pitts networks.

It is, however, clear by now that there are probably several other mechanisms as important or more important than the McCullogh-Pitts neuron. For instance, it is now well recognized that much processing takes place without somatic spikes, simply in terms of graded potentials. If graded signals play an important processing role, there must be *nonlinear* interactions between synaptic signals. The need for nonlinear operations that are more like multiplication than addition or subtraction has been customarily neglected by most neurophysiologists but is clearly critical for even the first stages of visual information processing.

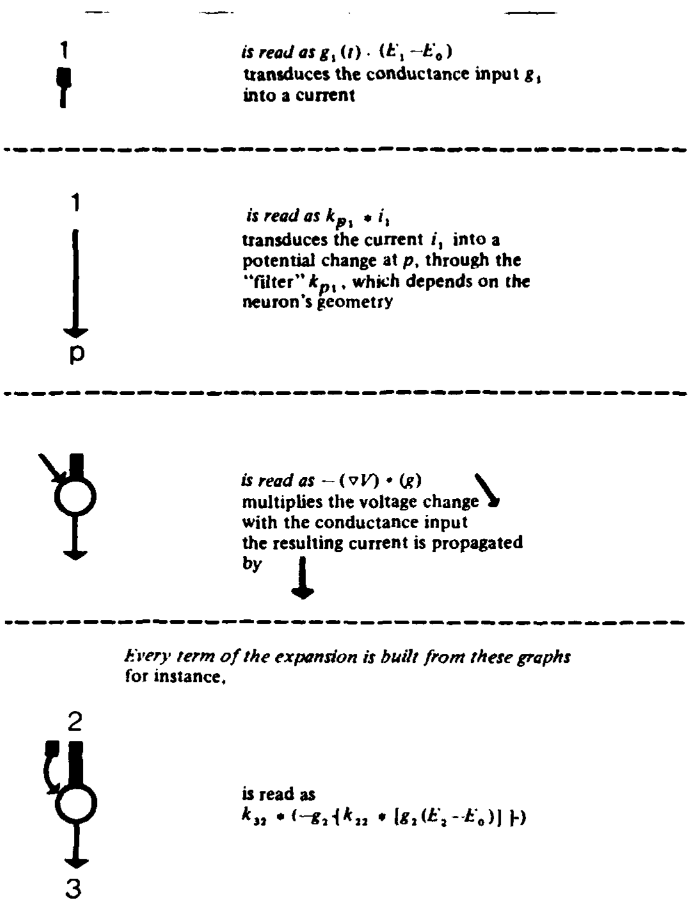
A simple biophysical mechanism that could underly nonlinear interactions between graded signals is already known. Since synaptic inputs are not current inputs but conductance changes to specific ions, synapses which are electrially close to each other on a cell's dendrite will mutually influence each other and result in a potential change at the soma which depends nonlinearly on the input signals. Probably the simplest and most common interaction of this type involves two synapses (or sets of synapses), one excitatory and the other inhibitory, increasing conductance to an ion with a battery close to the cell's resting potential. Activation of the inhibitory channel by itself will contribute nothing to the potential, but it may have a very powerful effect in shunting the potential towards the resting state when a neighbouring excitatory synapse becomes active. This shunting effect can be powerful and local. It can also be shown from the membrane's equations (Torre and Poggio, 1978; Poggio and Torre, 1981) that the interaction implemented is multiplication-like, of the type $g_1 - \alpha g_1 g_2$. This is in turn formally equivalent to an 'analogue' AND-NOT operation, one input (g_2) *vetoing* the other (g_1).

3.1. Synaptic interactions are polynomial functionals

The multiplication-like character of these synaptic interactions can be indeed demonstrated rigorously. An extension of cable theory shows that the voltage potential in an arbitrary dendritic structure is given by a specific Volterra series of the conductance inputs.

PROPOSITION 4 (Poggio & Torre, 1977)

The membrane potential in a passive dendritic tree is an entire functional for all bounded, transient conductance inputs.



is read as $g_1(t) \cdot (E_1 - E_0)$
transduces the conductance input g_1
into a current

is read as $k_{p_1} * i_1$
transduces the current i_1 into a
potential change at *p*, through the
"filter" k_{p_1}, which depends on the
neuron's geometry

is read as $-(\nabla V) \cdot (g)$ ↘
multiplies the voltage change
with the conductance input
the resulting current is propagated
by ↓

Every term of the expansion is built from these graphs
for instance,

is read as
$k_{32} * (-g_2\{k_{22} * [g_2(E_2 - E_0)]\})$

Figure 5. *The graphical notation for synaptic interactions*

Thus synaptic interactions of this type implement a specific type of the polynomial algorithms described earlier (their Volterra kernels have a specific structure). The polynomial structure of the interactions can be represented again by a graphic notation, where the various graphs give multiplication-like interactions of increasing degree and p-order between conductance changes (see fig.5).

This mathematical analysis begins to connect the morphology of the dendritic tree and the geometry of the synapses with the algorithms thereby implemented. Fig.6 shows several local circuits implementing simple nonlinear operations.

3.2. Dendritic Morphology and Information Processing

Koch et al.(1982) have proved that

PROPOSITION 5

In an arbitrary dendritic tree (without loops) the maximum veto effect is

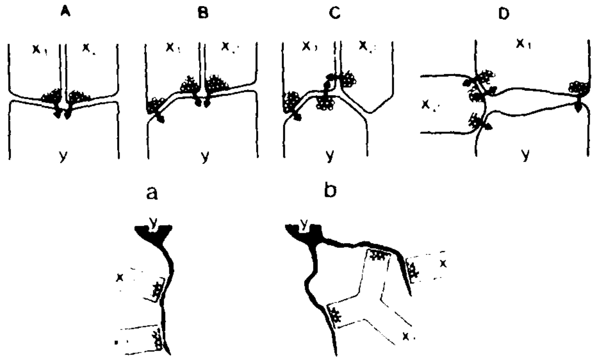


Figure 6. *Some local circuits performing different simple operations: a veto-like operation, a multiplication, a division, again a multiplication-like operation.*

obtained when (shunting) inhibition is on the direct path to the soma.

The role of the dendritic morphology in information processing has been studied (Koch et al., 1982; see also Poggio et al., 1981b and Poggio and Koch, 1981) in the case of retinal ganglion cells. In particular, we have examined with computer experiments on histological data the precise conditions underlying the effectiveness and the specificity of a *veto* interaction of the shunting type. The main result is that the effect can be powerful with physiological parameters values especially for dendritic morphologies of the δ type. In this case inhibition can veto specifically an excitatory input *if* it is on the *direct path* from the location of the excitatory synapse to the soma. As a consequence each class of cells may perform characteristic operations on their inputs depending on the branching and the geometry of the dendritic tree. Koch et al.(1982) have shown that δ and γ ganglion cells may underly different classes of logical-like operations because of their different branching patterns.

3.3. A Basic Elementary Mechanism

Because of the strength and specificity of such nonlinear interactions we have proposed that they may perform characteristic information processing operations in passive dendritic trees. Since inhibition *vetoes* effectively more distal excitatory inputs only when it is *on-path* to the source a variety of local operations can be performed, exploiting the branching geometry of a dendritic tree with a suitable localization of excitatory and inhibitory inputs. If this is true, a neuron would probably resemble an analogue LSI circuit with thousands of elementary processing units - the synapses - rather than a single logical gate. The idea that a veto- like operation plays an important role in visual information processing in the brain is not new, though its specific synaptic nature and properties probably are: Barlow has stressed many times, since his classical study of direction sensitive ganglion cells, (Barlow and Levick, 1965), that a veto-like operation is an important physiological mechanism in the retinal and cortical processes that underly perception. In the next section I will propose several simple visual algorithms—and corresponding neuronal circuits—that use this elementary veto mechanism.

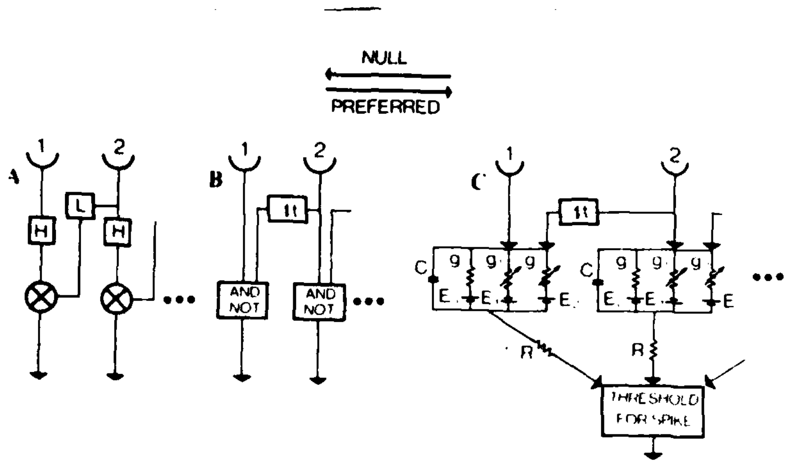


Figure 7. *Reichardt and Hassenstein's movement detector model, Barlow and Levick's scheme and Torre and Poggio's synaptic mechanism.*

4. THREE EXAMPLES OF NONLINEAR,LOCAL ALGORITHMS

4.1. Direction selective Motion Detection

The computation of local motion - in the simple sense of detecting the direction of motion - is a simple but fundamental step in machine and biological vision systems. It is straightforward to show that it is a nonlinear computation - i.e. linear operations alone could never be said to be truly direction selective. In terms of polynomial algorithms direction selectivity requires at least p-order 2 , i.e. a multiplication-like operation between 2 photoreceptors (or second order 'cells' linearly filtering the photoreceptor array). A linear system cannot give a time-averaged output that inverts sign for inversion of direction of motion, since $< L[x_1, x_2] >= L[< x_1 >, < x_2 >]$, if L is a linear mapping. A p-order 2 polynomial algorithm with a non-zero antisymmetric kernel component has the correct property. Thus:

PROPOSITION 6 (Poggio & Reichardt, 1976, 1981)

Direction selective motion detection - the average output must reverse sign for inversion of direction of motion - is p-order 2 (and degree 2).

In the specific case of the visual system of the fly there is convincing evidence that the p-order of the algorithm used is indeed 2 and not higher. The evidence rests on a variety of experiments briefly reviewed in (Poggio and Reichardt, 1976).

Thus the basic algorithm for direction selective motion detection is based on a multiplication-like interaction between pairs of inputs after asymmetric filtering (low-pass but also high-pass filters are possible). A particularly simple filtering operation is an asymmetric delay (see fig. 7). It is easy to prove that correlation models are equivalent to p-order 2 polynomial algorithms:

PROPOSITION 7 (Poggio & Reichardt, 1976)

Correlation models of motion detectors (in the sense of Reichardt) are a subclass of antisymmetric, p-order 2, polynomial algorithms.

How can this algorithm be implemented in neural hardware? If we follow the ideas outlined earlier, the obvious choice would be to use a synaptic mechanism of the veto type at the level of a cell's dendrite. As shown in fig. 7, Barlow and Levick had in fact proposed from their physiological experiments on directional selective ganglion cells, an AND-NOT operation as the basis for directional selectivity to motion. Torre and Poggio (1978) conjectured that the synaptic veto effect described earlier may be the mechanism whereby directional selectivity is achieved. Provided that suitable conditions on the ionic channels, the geometry of the dendritic tree and the localization of synapses are satisfied, their conjecture certainly fits two of the main experimental properties of direction selective cells, namely that the interactions responsible are between local subunits of the receptive field, and that they are inhibitory. Thus the basic algorithm used for simple motion detection in various biological systems may indeed by based on the synaptic veto mechanism. In particular, Koch et al. (1982) have recently proposed that a δ-cell-like morphology is the substratum of direction selectivity in the retina of the cat.

4.2. Detection of Relative Movement

Discontinuities in the optical flow field—the distribution of apparent velocities on the eyes—are a good indication of object boundaries and can be used to segment images into regions that correspond to different objects. In particular, the relative motion of an object against a textured background can be used to reveal its presence and to delineate its boundaries. The human visual system is very efficient at this task. Quite similarly a fly is able to detect and discriminate an object that moves relative to a ground texture.

In terms of polynomial algorithms this computation is of p-order 4 (although p-order 2, degree 4 may also be sufficient in specific cases, see Reichardt and Poggio, 1979). Many experiments have established that the fly indeed uses an algorithm which is mainly p-order 4 (it has also higher terms). More precisely the behavioral data which measure the fixation response of the fly to a textured small figure oscillating sinusoidally with various phases in front of an oscillating ground texture - shows that the basic algorithm relies on an inhibitory multiplication - like operation between motion detector units (Reichardt and Poggio, 1979).

Again the synaptic veto mechanism of shunting inhibition seems an ideal candidate for implementing this operation. The overall circuitry is shown in fig. 8 (Poggio et al., 1981a). Large field cells summate the output of many elementary motion detectors and inhibit via presynaptic shunting inhibition the single elementary motion detectors. This circuitry accounts well for a large body of existing behavioral experiments; many more predictions have been successfully

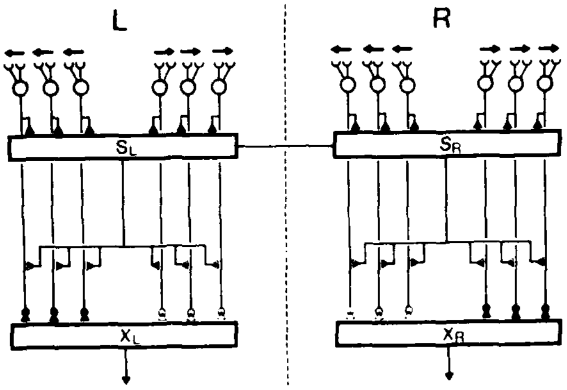


Figure 8. *The circuit (and algorithm thereby implemented) possibly used by the visual system of the fly to perform the detection of discontinuities in the optical flow. Redrawn from Poggio et al., 1981*

tested. In particular, the dynamics of the fly's behavioral response is quantitatively predicted by this algorithm for a variety of figure-ground descrimination tasks. Presynaptic inhibition with an equilibrium potentical near the resting potential is conjectured to implement the key operation in the algorithm, which amounts to a comparison of large field motion with local measurements. This circuitry, considered as an algorithm for detecting discontinuities in the optical flow (it has p-order 4 and higher), is efficient and reliable, as shown by several computer experiments on textured patterns.

4.3. The Detection of Zero-crossings

Over the past twenty years researchers in computer vision have proposed several algorithms to detect and represent various kinds of intensity changes. I will focus here on one of them because of its potential implications for cortical processing. The basic ideas were suggested to D. Marr and myself (Marr and Poggio, 1977, 1979), while working on the problem of human stereo, from a combination of psychophysical data (by H. Wilson) and of recent results in the field of complex analysis . Briefly the scheme consists of filtering the image through a number of independent bandpass operations that simultaneously blur and take a second spatial derivative. Changes in intensity are then localized separately in each of the filtered versions of the image by detecting the loci of zero values, i.e. the zero-crossings. Zero-crossings are a close relative of physical edges and can be used for later processing; they are for instance used in the stereo algorithm developed at MIT (Grimson, 1981) and elsewhere. To help in understanding why zero-crossings in bandpass channels may be useful discrete symbols to extract, I will describe a result in complex analysis that I still find intriguing and fascinating. In 1977 B. Logan (1977) proved that under some technical conditions an appropriately bandpass signal can be completely reconstructed from its zero-crossings alone. A successful extension of this theorem to images by Nishihara and Poggio (Poggio, 1982; Poggio et al., 1982), though not strictly valid for the channels described by H. Wilson, suggests that zero-crossings are very rich in information about the filtered image. Ideas based on Logan's type of results are attractive especially from the point of view of visual psychophysics and physiology, since they seem to provide a theoretical basis for the existence of edge detectors in the output of bandpass channels in the visual system, thus providing a potential synthesis of the edge detectors ideas with the frequency channels evidence. Marr and Hildreth (1980) have provided a number of attractive heuristic arguments for justifying a slight variation of the original scheme (Marr and Poggio, 1977). In particular they proposed that the initial filtering of the image was performed by nondirectional (as opposed to oriented) receptive fields, again described as differences of gaussians (DOG) (which approximates the operation of taking the Laplacian of the image filtered through a gaussian, see Appendix.) Since X retinal ganglion cells have a DOG receptive field and are usually described as linear filters, it is not too unreasonable to propose that the filtering operation is indeed performed in the retina and represented by the activity in the ON and OFF layers of ganglion cells, positive values being represented by ON center X cells and negative values by OFF center X cells. Thus the binary map of the convolved image shown in fig. 9 would represent the combined map of activity in the OFF and ON layers of ganglion cells in the retina.

How can the zero-crossings - the transition of activity between ON and OFF cells - be detected?

Fig. 10 shows that a mechanism connecting neighboring ON and OFF cells with an AND gate, possibly implemented via synaptic mechanisms of the Poggio-Torre type (with a shunting conductance decreasing input, see Koch et al., 1982), could detect zero-crossing lying between the two rows of cells. This scheme, proposed by Marr and Hildreth (1980) does not require the inhibition which seems to be involved in the main properties of cortical cells, like orientation and direction selectivity. An alternative scheme can, however, be based on the synaptic veto mechanism.

The critical observation is that a zero-crossing is also defined by activity in the ON layer and *absence* of activity in neighboring ON cells (and conversely for the OFF layer). Thus a zero-crossing can be detected by avoidance of inhibition, logically equivalent to an AND-NOT operation. It is a simple matter to adapt this idea to create an oriented zero-crossing segment detector as shown in fig. 10. Since the veto operation can be performed by distal excitation (on spines ?) and inhibition of the shunting type on the proximal part of a single dendrite, the same cell may perform independently this operation on the OFF and on the ON layer on different dendrites, adding the two results for increasing reliability. Interestingly, however, either the ON or the OFF layer alone are sufficient. Notice that in a standard map of the receptive field inhibition may be invisible and only excitatory inputs from ON and OFF cells (on different dendrites) may be measureable (and linear). In this scheme unbalanced receptive fields (loosely corresponding to sustained properties) are not only advantageous but probably (as suggested by K. Richter) necessary for a robust physiological implementation. A trivial property of the circuitry follows from the preceeding sections:

COROLLARY

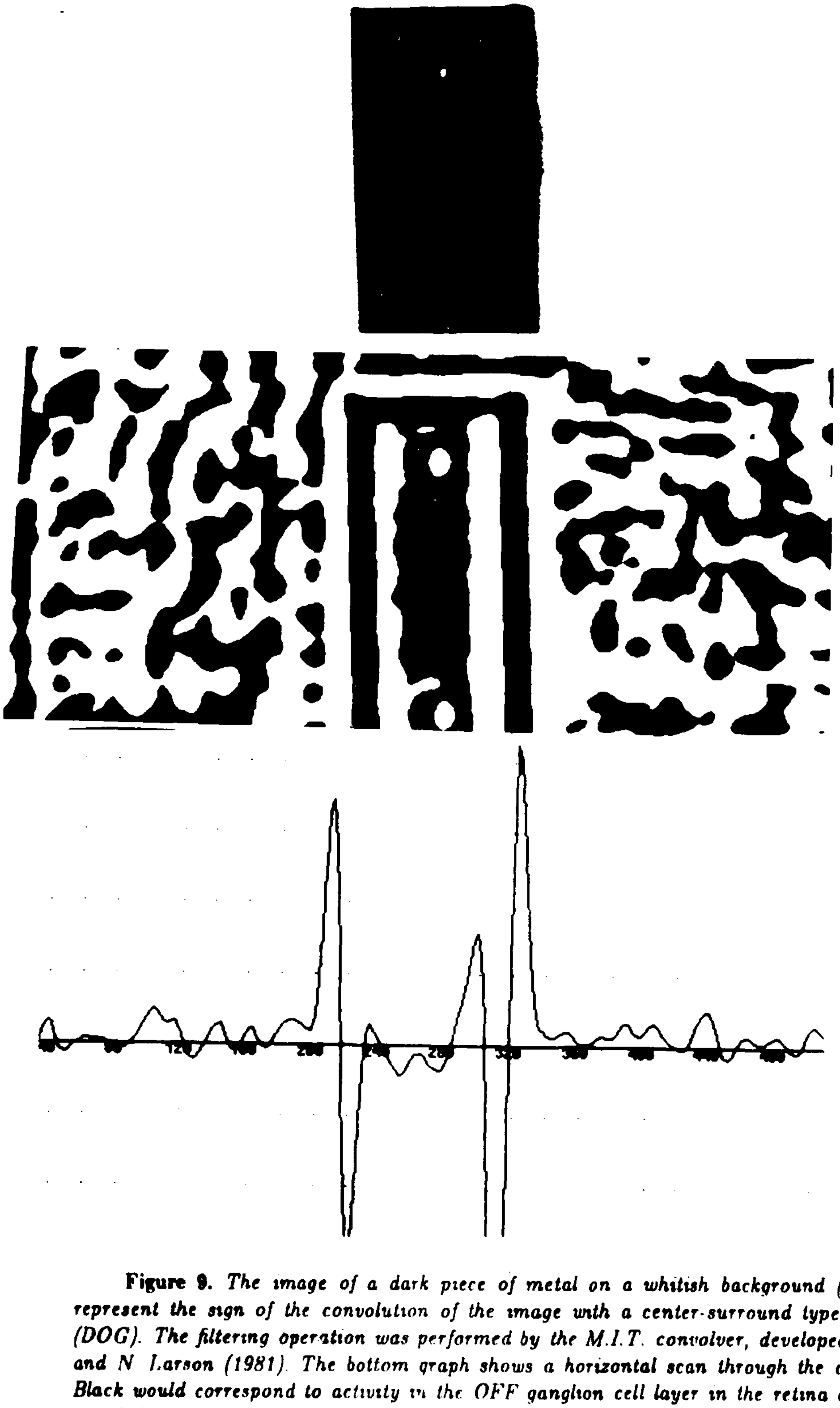


Figure 9. *The image of a dark piece of metal on a whitish background (top). The middle represent the sign of the convolution of the image with a center-surround type of receptive field (DOG). The filtering operation was performed by the M.I.T. convolver, developed by K. Nishihara and N Larson (1981). The bottom graph shows a horizontal scan through the convolution array. Black would correspond to activity in the OFF ganglion cell layer in the retina and no activity in the ON ganglion cells, while white would correspond to the complementary activity pattern.*

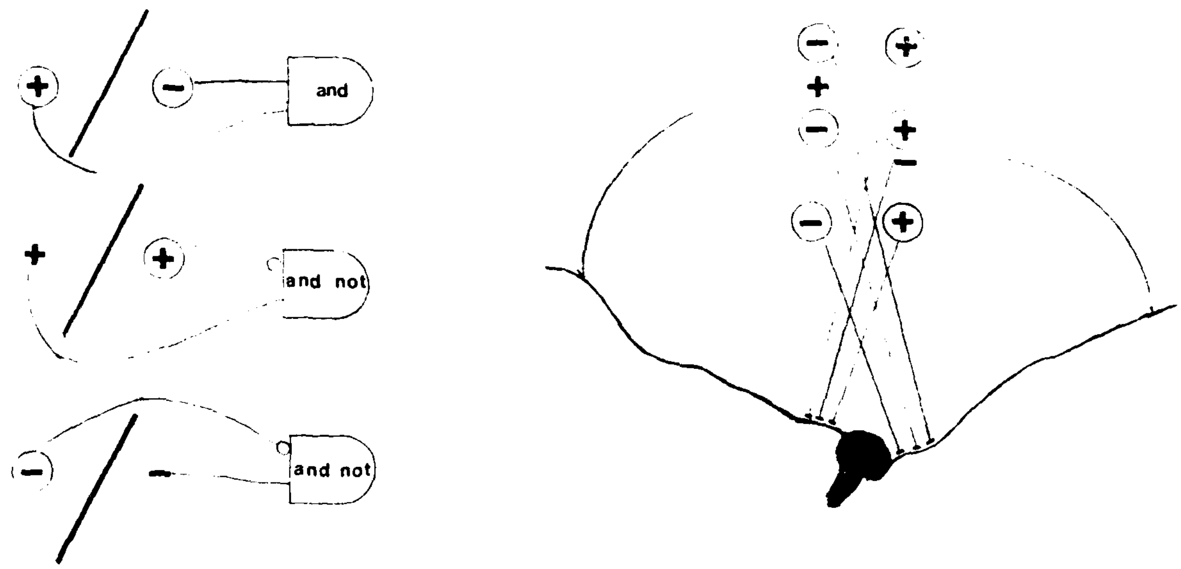


Figure 10. *Various models of zero-crossing detectors. In (a) a zero-crossing corresponding to transition of activity between the ON layer and the OFF layer (like the zero-crossing corresponding to the first edge from the right in fig. 9) can be detected by an AND mechanism between adjacent regions of the ON and the OFF layer, as proposed by Marr and Hildreth (1980): either input alone is "invisible". In (b) and (c) the same zero-crossing is detected by an AND-NOT operation on either the ON or the OFF layer (circled inputs are "invisible", unless the other excitatory input is simultaneously active, as it is the case for so-called "silent" inhibition). All these operations may be performed by nonlinear, p-ordebr 2 interactions. A cell summating linearly these last two operations, performed on different dendrites, is sketched on the right side of the figure.*

A veto-like zero-crossing detector can be regarded as a p-order 2 polynomial algorithm on the ganglion cells array.

With an appropriate transformation of the input its degree can be as low as degree 2 (see earlier example). Thus this way of detecting zero-crossings is equivalent to taking measurements on the ganglion cell activity that are degree 2, p-order 2 (nonlinear) functionals.

This idea can be easily extended to account for directional selective properties, of some cortical cells. The first possibility is to gate the schemes of fig.10 (left) with an hypothetically transient Y-cell input. The resulting algorithm would be similar to the scheme proposed by Marr and Ullman (1981), where all AND operations would be substituted with AND-NOT operations in the way suggested earlier. Another possibility is a scheme similar to the models of fig.7. As in the fly motion detector scheme, a low-pass operation on one of the two channels (or high-pass on the other) endows the zero-crossing detector scheme with direction selective properties (see fig. 11). Since precision is needed in the detection of the zero-crossing, the low-pass

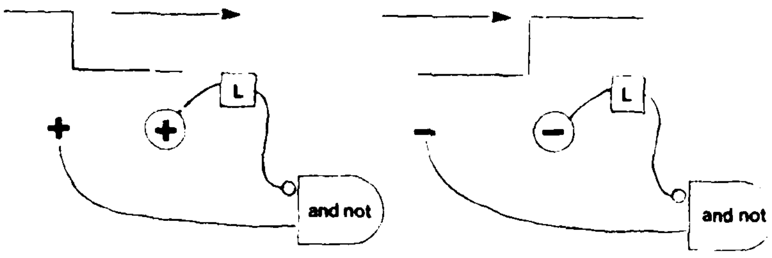


Figure 11. *A prediction moving zero-crossings are detected - with directional selective properties - by the schemes sketched here: light edges by the ON mechanism, dark edges by the OFF mechanism. If a cell has to detect the same physical edge moving in both directions, then it may use both the ON and the OFF mechanism - on different dendrites.*

ɪt must operate on the inhibitory input. It follows then that light edges can be
.ed only by the ON system and dark edges by the OFF system (if directional
vity is required: otherwise there is no such restriction). This prediction may
ɔported by recent pharmacological experiments of P. Schiller. Interestingly,
lgorithm is again very similar to the fly's (and Barlow's) movement detector,
ting on a specific linear transformation of the retinal image (DOGs), instead of
ual gaussians induced by the optics (the ρ point-spread functions of Geiger and
o (1975) are actually *general* linear transformations of the retinal image) Thus,
n detection in insects may indeed be very similar to the detection of moving
ɪon-oriented !) zero-crossings, since center-surround filtering is known to occur
: motion detection in the fly's visual pathway! Similar very simple schemes,
ɪased on the synaptic veto mechanism, seem capable of accounting for several
rties of cortical binocular cells. Keith Nishihara has actually developed similar
ɪes in a fast stereo algorithm for robotics applications recently implemented
e Lisp machine.

References

Barlow, H.B. Critical limiting factors in the design of the eye and visual cortex. Proc. R. Soc. Lond. B 212, 1-34 (1981)

Barlow, H.B., Levick, W.R. The mechanism of directionally selective units in rabbit's retina. J. Physiol. 178, 477-504 (1965)

Geiger, G., Poggio, T. The Orientation of Flies towards Visual Patterns: On the Search for the Underlying Functional Interactions. *Biol. Cyber.*, **19**, 39-54, (1975).

Grimson, W.E.L.: From images to surfaces. M.I.T. Press (1981)

Koch, C., Poggio, T. and Torre, V., Retinal Ganglion Cells: A Functional Interpretation of Dendritic Morphology, Phil. Trans. R. Soc.B, 298, 227-264, (1982)

Kohonen, T.: Associative memory. A system-theoretical approach. Berlin, Heidelberg, New York: Springer (1977)

Kolmogorov, A.N.: On the representation of continuous functions of several variables by superpositions of continuous functions of one variable and addition (in russian). Dokl. Akad. Nauk SSSR 114, 679-681 (1957); Am. Math. Soc. Transl. 28, 55-59 (1963)

Logan Jr., B. F. Information in the Zero-crossings of Bandpass Signals, The Bell System Technical Journal, Vol. 56, 487-510 (1977)

Marr, D.: Vision. W.H. Freeman, San Francisco (1982)

Marr, D., Hildreth, E. 1980 Theory of Edge Detection Proc. Roy. Soc. Lond. B 207, 187-217 (1980)

Marr, D. C., Poggio, T. From Understanding Computation to Understanding Neural Circuitry. In: MECHANISMS IN VISUAL PERCEPTION, E. Poppel, R. Held and J. E. Dowling, eds. *Neurosciences Res. Prog. Bull.*, **15**, No. 3, pp. 470-488, 1977 .Also available as *A. I. Memo 357*, Massachusetts Institute of Technology Artificial Intelligence May 1976.

Marr, D., Poggio, T. A Theory of Human Stereo Vision. *A. I. Memo 451*, November 1977, Massachusetts Institute of Technology, Artificial Intelligence Laboratory. Also avalable in a shorter version as A Computational Theory of Human Stereo Vision. *Proc. R. Soc. Lond. B*, **204**, 301-328, (1979.)

Marr, D., Ullman, S.: Directional selectivity and its use in early visual processing. Proc. R. Soc. Lond. B 211, 151-180, (1981)

Minsky, M., Papert, S.: Perceptrons. An introduction to computational geometry. Cambridge. Mass., London: MIT Press (1969)

Nishihara, H.K.: Intensity, Visible-Surface, and Volumetric Representations. Artif. Intell. 17, 265-284 (1981)

Nishihara, H.K., Larson, N.G. Proc. DARPA image Understanding Workshop 114-120 Science Application Inc., Arlington, VA (1981)

Palm, G.: On representation and approximation of nonlinear systems. Biol. Cybernetics 31, 119-124 (1978)

Palm, G.: On representation and approximation of nonlinear systems. Part II. Biol. Cybernetics 34, 49-52 (1979)

Palm, G., Poggio, T. The Volterra Representation and the Wiener Expansion: Validity and Pitfalls. *SIAM J. Appl. Math*, **33**, 195-216, (1977)

Palm, G., Poggio, T. Stochastic Identification Methods for Nonlinear Systems: An Extension of the Wiener Theory. *SIAM J. Appl. Math.*, **34**, 524-534, (1978)

Poggio, T. On Optimal Nonlinear Associative Recall. *Biol. Cyber.*, **19**, 201-209, (1975a)

Poggio, T. On Optimal Discrete Estimation. PROCEEDINGS OF THE FIRST SYMPOSIUM ON TESTING AND IDENTIFICATION OF NONLINEAR SYSTEMS, March 17-20, 1975, California Institute of Technology, Pasadena, California. G. D. McCann and P. Z. Marmarelis, eds., pp. 30-37 (1975b)

Poggio, T., Marr's Computational Approach to Vision, Trends in NeuroSciences - October, Vol. 4, No.10, pp.258-262, (1981)

Poggio, T., Trigger Features or Fourier Analysis in Early Vision: A New Point of View, In: The Recognition of Pattern and Form, ed D. Albrecht, Springer, (1982)

Poggio, T. and Koch, C. Nonlinear Interactions in a Dendritic Tree: Localization, Timing and Role in Information Processing, A. I. Memo No.657, September 1981, Massachusetts Institute of Technology Artificial Intelligence Laboratory (1981)

Poggio, T., Koch, C. and Torre, V., Microelectronics in Nerve Cells: Dendritic Morphology and Information Processing, A. I. Memo No.650, October 1981, Massachusetts Institute of Technology Artificial Intelligence Laboratory (1981b)

Poggio, T., Nishihara, H.K. and Nielsen, K.R.K. Zero-crossings and Spatiotemporal Interpolation in Vision: Aliasing and electrical coupling between sensors, A.I. Memo No. 675, May 1982, Massachusetts Institute of Technology Artificial Intelligence Laboratory (1982)

Poggio, T., Reichardt, W. Visual Control of Orientation Behaviour in the Fly. Part II. Towards the Underlying Neural Interactions. *Quart. Rev. Biophys.*, **9**, 3, 377-438, (1976)

Poggio, T., Reichardt, W. On the Representation of Multi-Input Systems: Computational Properties of Polynomial Algorithms. *Biol. Cyber.*, **37**, 167-186, (1980)

Poggio, T., Reichardt, W. Characterization of Nonlinear Interactions in the Fly's Visual System. pp. 64-84, Appendix 4: A Polynomial Representation of Algorithms, pp. 197-202. In: THEORETICAL APPROACHES IN NEUROBIOLOGY, W. Reichardt and T. Poggio, eds., MIT Press, Cambridge, Mass., (1981)

Poggio, T., Reichardt, W., Hausen, K. A Neuronal Circuitry for Relative Movement Discrimination by the Visual System of the Fly. *Naturwissenschaften*, **68**, 443-466, (1981a)

Poggio, T. and Rosser, B., The Computational Problem of Motor Control, In: Machine Intelligence 10, ed's Hayes, Michie, Pao (1982)

Poggio, T., Torre, V. A new approach to synaptic interaction. In: Lecture Notes in Biomathematics 21. Theoretical Approaches to Complex Systems, Proceedings, Tubingen, June 11-12, 1977. Eds. R. Heim and G. Palm Springer-Verlag, Berlin Heidelberg New York, pp 89-115 (1977)

Poggio, T., Torre, V. A theory of synaptic interactions. In: Theoretical approaches in neurobiology. Eds. W.Reichardt and T.Poggio. M.I.T. Press (1981)

Reichardt, W., Poggio, T. Figure-Ground Discrimination by Relative Movement in the Visual System of the Fly. Part I: Experimental Results. *Biol. Cyber.*, **35**, 81-100, (1979)

Resnikoff, H.L., On the Psychophysical function, J. Math Biol. 2, 295-276, (1975)

Torre, V., Poggio, T. A Synaptic Mechanism Possibly Underlying Directional Selectivity to Motion. *Proc. R. Soc. Lond. B*, **202**, 409-216, (1978)

APPENDIX

V. Torre and T. Poggio

Instead of the Laplacian of a gaussian as the underlying filter, it is appealing to consider the second directional derivative along the gradient of the image filtered through a gaussian and consider its zero-crossings. The second directional derivative along the gradient has the form (in cartesian coordinates)

$$\frac{f_x^2 f_{xx} + 2 f_x f_y f_{xy} + f_y^2 f_{yy}}{f_x^2 + f_y^2}$$

to be compared with the Laplacian

$$f_{xx} + f_{yy}$$

where $f(x,y)$ represents the image convolved with a gaussian point spread function. The first operator is nonlinear and symmetric. It reduces to the Laplacian for "one-dimensional" patterns f depending only on one spatial variable. In addition, the second directional derivative of a (symmetric) gaussian along the gradient is quite similar to the Laplacian of a gaussian Thus, for circularly symmetric patterns filtered through a gaussian, the two operators lead to very similar results. Thus, the two operators cannot be distinguished in physiological experiments using either fully circularly symmetric or one-dimensional patterns like gratings or bars. It has already been observed by several authors (for instance, J. Canny, per. comm.) that the second directional derivative along the gradient appears to be a better and more natural operator for edge detection than the Laplacian (see Torre and Poggio, in prep.).